IN THE CLAIMS

Claim 1 (currently amended): A logic circuit comprising:

data flow logic; ~~and~~

control flow logic to select and fetch a trace descriptor for processing, the fetched trace descriptor including at least one dependency descriptor, the control flow logic to dispatch to the data flow logic a dependency descriptor including dependency information having live-in information and live-out information for an instruction sequence and an address of the instruction sequence;

the data flow logic coupled to the control flow logic to receive the dispatched dependency descriptor, to fetch the instruction sequence using the address from the received dependency descriptor, and to execute the instruction sequence according to the dependency information in the received dependency descriptor[[.]]; and

an issue window coupled between the control flow logic and the data flow logic, the issue window to store the dependency descriptor dispatched from the control flow logic wherein the issue window includes a first portion to store active dependency descriptors and a second portion to store inactive dependency descriptors, wherein an inactive dependency descriptor is to remain in the second portion until data to be used by the corresponding instruction sequence is available.

Claim 2 (previously presented): The logic circuit of claim 1 comprising a storage area coupled to the control flow logic and the data flow logic, the storage area to store the dependency descriptor dispatched by the control flow logic.

Claim 3 (previously presented): The logic circuit of claim 1 comprising a storage area coupled to the control flow logic, the storage area to store a trace descriptors.

Claim 4 (previously presented): The logic circuit of claim 3 comprising a second storage area coupled to the data flow logic, the second storage area to store instructions contiguously based on dependency information.

Claim 5 (previously presented): The logic circuit of claim 1 comprising a storage area coupled to the data flow logic and control flow logic, the storage area to store live-out data.

Claim 6 (previously presented): The logic circuit of claim 1 comprising a storage area coupled to the control flow logic, the storage area to map dependency information.

Claim 7 (canceled).

Claim 8 (previously presented): The logic circuit of claim 1 wherein the trace descriptor includes aggregate live-in information for a plurality of dependency descriptors in the trace descriptor.

Claim 9 (previously presented): The logic circuit of claim 1 wherein the trace descriptor includes aggregate live-out information for a plurality of dependency descriptors in the trace descriptor.

Claim 10 (currently amended): A computer system comprising:

at least one memory device to store trace descriptors and instruction sequences, each trace descriptor associated with a trace;

a bus coupled to the at least one memory device;

control flow logic to select and fetch one of the trace descriptors, the fetched trace descriptor including ~~aggregate live-in information and aggregate live-out information for the corresponding trace,~~ a plurality of dependency descriptors having locations of corresponding instruction sequences within the trace and having dependency information for the corresponding instruction sequences; ~~and~~

data flow logic coupled to the control flow logic to receive a dependency descriptor dispatched from the control flow logic, to fetch an instruction sequence corresponding to the received dependency descriptor, and to execute the fetched instruction sequence according to dependency information in the received dependency descriptor[[.]]; and

an issue window coupled between the control flow logic and the data flow logic, the issue window to store the dependency descriptor dispatched from the control flow logic wherein the issue window includes a first portion to store active dependency descriptors and a second portion to store inactive dependency descriptors, wherein an inactive dependency descriptor is to remain in the second portion until data to be used by the corresponding instruction sequence is available.

Claims 11 and 12 (canceled).

Claim 13 (previously presented): The computer system of claim 10 wherein the at least one memory device is to store an instruction sequence contiguously based on dependency information.

Claim 14 (previously presented): The computer system of claim 10 comprising a storage area coupled to the data flow logic and control flow logic, the storage area to store live-out data.

Claim 15 (previously presented): The computer system of claim 10 comprising a storage area coupled to the control flow logic, the storage area to map dependency information.

Claims 16-18 (canceled).

Claim 19 (previously presented): The computer system of claim 10 wherein dependency information in the received dependency descriptor includes live-in and live-out information.

Claim 20 (previously presented): A method of processing instructions comprising:

selecting and fetching a trace descriptor from a trace storage area in accordance with program control flow;

identifying from the fetched trace descriptor a dependency descriptor including dependency information for a set of instructions and an address of the set of instructions;

dispatching the dependency descriptor for execution;

fetching the set of instructions from an instruction storage separate from the trace storage area using the address from the dispatched dependency descriptor; and

executing the set of instructions according to dependency information in the dispatched dependency descriptor.

Claim 21 (previously presented): A method according to claim 20 comprising:

updating live-out data in a first storage area.

Claim 22 (previously presented): A method according to claim 20 comprising:

storing the dependency descriptor from control flow logic into a storage area; and

reading the dependency descriptor out of the storage area into the data flow logic.

Claim 23 (previously presented): A method according to claim 20 wherein the fetching of the set of instructions is completed just in time for execution.

Claim 24 (previously presented): A method according to claim 20 wherein the executing comprises executing instructions out of order.

Claim 25 (previously presented): A method according to claim 21 comprising:

updating an architectural state using the data in the storage area.

Claim 26 (previously presented): A method according to claim 25 comprising:

recovering an earlier architectural state after a misprediction using the data in the storage area.

Claim 27 (previously presented): A method according to claim 20 wherein the selecting comprises predicting the next trace descriptor to process.

Claim 28 (previously presented): A machine-readable medium that provides instructions, which when executed by a machine cause the machine to perform operations comprising:

selecting and fetching a trace descriptor in accordance with program control flow;

identifying from the fetched trace descriptor a dependency descriptor including dependency information for a set of instructions of a dependency chain and an address of the set of instructions, the dependency information to indicate at least one data on which the dependency chain depends;

dispatching the dependency descriptor for execution;

fetching the set of instructions using the address from the dispatched dependency descriptor; and

executing the set of instructions according to dependency information in the dispatched dependency descriptor.

Claim 29 (previously presented): The machine-readable medium of claim 28, wherein the operations comprise:

updating live-out data in a first storage area.

Claim 30 (previously presented): The machine-readable medium of claim 28, wherein the operations comprise:

storing the dependency descriptor in an issue window by control flow logic; and

reading the dependency descriptor out of the issue window into the data flow logic.

Claim 31 (previously presented): The logic circuit of claim 1 wherein the fetched trace descriptor includes a plurality of dependency descriptors having addresses of corresponding instruction sequences and having dependency information for corresponding instruction sequences.

Claims 32–33 (canceled).

Claim 34 (previously presented): The computer system of claim 10, wherein the data flow logic includes a plurality of clusters each to independently execute different fetched instruction sequences each corresponding to a different received dependency descriptor.

Claim 35 (canceled).

Claim 36 (currently amended): The system of claim 10, wherein only results of the execution corresponding to the aggregate live-out information are to be globally ~~broadcast~~ broadcasted.